

#2
PATENT APPLICATION

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re the Application of

Masaki KYOJIMA and Kil-ho SHIN

Application No.: New U.S. Application

Filed: June 6, 2000

Docket No.: 106406

For: DATA GENERATING APPARATUS AND DATA VERIFYING APPARATUS



CLAIM FOR PRIORITY

Director of the U.S. Patent and Trademark Office
Washington, D.C. 20231

Sir:

The benefit of the filing date of the following prior foreign application filed in the following foreign country is hereby requested for the above-identified patent application and the priority provided in 35 U.S.C. §119 is hereby claimed:

Japanese Patent Application No. 11-226380 filed August 10, 1999.

In support of this claim, a certified copy of said original foreign application:

 X is filed herewith.

 was filed on in Parent Application No. filed .

It is requested that the file of this application be marked to indicate that the requirements of 35 U.S.C. §119 have been fulfilled and that the Patent and Trademark Office kindly acknowledge receipt of this document.

Respectfully submitted,

James A. Oliff
Registration No. 27,075

JAO:TJP/emb

Thomas J. Pardini
Registration No. 30,411

Date: June 6, 2000

OLIFF & BERRIDGE, PLC
P.O. Box 19928
Alexandria, Virginia 22320
Telephone: (703) 836-6400

<p>DEPOSIT ACCOUNT USE AUTHORIZATION Please grant any extension necessary for entry; Charge any fee due to our Deposit Account No. 15-0461</p>

日 本 国 特 許 庁
PATENT OFFICE
JAPANESE GOVERNMENT

JC856 U.S. PTO
09/588049
06/06/00

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出 願 年 月 日

Date of Application:

1999年 8月10日

出 願 番 号

Application Number:

平成11年特許願第226380号

出 願 人

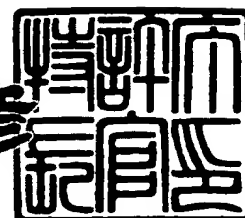
Applicant(s):

富士ゼロックス株式会社

2000年 4月21日

特許庁長官
Commissioner,
Patent Office

近 藤 隆 彦



出証番号 出証特2000-3028010

【書類名】 特許願

【整理番号】 FN99-00075

【提出日】 平成11年 8月10日

【あて先】 特許庁長官殿

【国際特許分類】 H04L 9/30

【発明の名称】 データ生成装置およびデータ検証装置ならびにその方法

【請求項の数】 32

【発明者】

【住所又は居所】 神奈川県足柄上郡中井町境 4 3 0 グリーンテクなかい
富士ゼロックス株式会社内

【氏名】 京嶋 仁樹

【発明者】

【住所又は居所】 神奈川県足柄上郡中井町境 4 3 0 グリーンテクなかい
富士ゼロックス株式会社内

【氏名】 申 吉浩

【特許出願人】

【識別番号】 000005496

【氏名又は名称】 富士ゼロックス株式会社

【電話番号】 0462-38-8516

【代理人】

【識別番号】 100086531

【弁理士】

【氏名又は名称】 澤田 俊夫

【電話番号】 03-5541-7577

【手数料の表示】

【予納台帳番号】 038818

【納付金額】 21,000円

【提出物件の目録】

【物件名】 明細書 1

【物件名】	図面	1
【物件名】	要約書	1
【プルーフの要否】	要	

【書類名】 明細書

【発明の名称】 データ生成装置およびデータ検証装置ならびにその方法

【特許請求の範囲】

【請求項 1】 第 1 のデータを保持する主検査対象データ記憶部と、
第 2 のデータを保持する副検査対象データ記憶部と、
上記副検査対象データ記憶部に記憶されたデータから暗号鍵を作成する暗号鍵生成部と、

上記主検査対象データ記憶部に記憶されたデータを上記暗号鍵生成部が生成した暗号鍵で暗号化する暗号部とを備え、

上記暗号部で暗号化された結果および上記副検査対象データ記憶部に記憶されたデータの少なくとも一方を含むデータを作成することを特徴とするデータ生成装置。

【請求項 2】 前鍵を保持する前鍵保持部を更に備え、上記暗号鍵生成部が暗号鍵を生成する際に、上記前鍵保持部に記憶された前鍵も使用することを特徴とする請求項 1 記載のデータ生成装置。

【請求項 3】 上記主検査対象データ記憶部に記憶されるデータが、所定の暗号文を復号した結果であることを特徴とする請求項 1 記載のデータ生成装置。

【請求項 4】 上記主検査対象データ記憶部に記憶されるデータが、所定のデータの署名値であることを特徴とする請求項 1 記載のデータ生成装置。

【請求項 5】 上記暗号鍵生成部が一方向性関数で構成されており、該一方向性関数に上記副検査対象データ記憶部に記憶されたデータを入力して得られた結果が上記暗号鍵である請求項 1 記載のデータ生成装置。

【請求項 6】 上記前鍵を暗号化するための前鍵暗号鍵を保持する前鍵暗号鍵記憶部と、前鍵暗号鍵記憶部に記憶された鍵で前鍵を暗号化する前鍵暗号部とを備えることを特徴とする請求項 2 記載のデータ生成装置。

【請求項 7】 上記暗号部で行われる暗号化が、共通鍵暗号であることを特徴とする請求項 1 記載のデータ生成装置。

【請求項 8】 上記暗号部で行われる暗号化が、所定の法数のもとでの上記主検査対象データ記憶部に記憶されているデータと、上記暗号鍵生成部で生成さ

れた暗号鍵の乗除演算であることを特徴とする請求項 1 記載のデータ生成装置。

【請求項 9】 第 1 のデータから暗号鍵を作成するステップと、
所定の特徴を持つことが検査可能な第 2 のデータを該暗号鍵で暗号化するステップと、

第 1 のデータおよび暗号化された第 2 のデータの少なくとも一方を含むデータを作成するステップとからなるデータ生成方法。

【請求項 1 0】 第 1 のデータを保持する主検査対象データ記憶部と、
第 2 のデータを保持する副検査対象データ記憶部と、
上記副検査対象データ記憶部に記憶されたデータから復号鍵を作成する復号鍵生成部と、

上記主検査対象データ記憶部に記憶されたデータを上記復号鍵生成部が生成した復号鍵で復号する復号部と、

上記復号部で復号されたデータが所定の特徴を持つかどうかを検査する検査部とを備えることを特徴とするデータ検証装置。

【請求項 1 1】 前鍵を保持する前鍵保持部を更に備え、上記復号鍵生成部が復号鍵を生成する際に、上記前鍵保持部に記憶された前鍵も使用することを特徴とする請求項 1 0 記載のデータ検証装置。

【請求項 1 2】 上記検査部が、上記復号部で復号されたデータが所定のデータを所定の復号鍵で復号した結果であることを検査することを特徴とする請求項第 1 0 記載のデータ検証装置。

【請求項 1 3】 上記検査部が、上記復号部で復号されたデータが所定のデータを所定の署名鍵で署名した署名値であることを検査することを特徴とする請求項 1 0 記載のデータ検証装置。

【請求項 1 4】 上記復号鍵生成部が一方向性関数で構成されており、該一方向性関数に上記副検査対象データ記憶部に記憶されたデータを入力して得られた結果が上記復号鍵である請求項 1 0 記載のデータ検証装置。

【請求項 1 5】 暗号化された前鍵を記憶する前鍵記憶部と、
暗号化された上記前鍵を復号する復号鍵を記憶する前鍵復号鍵記憶部と、
上記前鍵復号鍵記憶部に記憶された上記復号鍵で上記前鍵記憶部に記憶された

暗号化された前鍵を復号する前鍵復号部とをさらに備えることを特徴とする請求項 1 0 記載のデータ検証装置。

【請求項 1 6】 上記復号部で行われる復号が、共通鍵暗号の復号であることを特徴とする請求項 1 0 記載のデータ検証装置。

【請求項 1 7】 上記復号部で行われる復号が、所定の法数のもとでの暗号化された被検査値と復号鍵の乗除算であることを特徴とする請求項 1 0 記載のデータ検証装置。

【請求項 1 8】 第 1 のデータから復号鍵を作成するステップと、
第 2 のデータを該復号鍵で復号するステップと、
復号の結果が所定の特徴を持つことを検査するステップとからなるデータ検証方法。

【請求項 1 9】 データ生成装置と、該生成装置で生成されたデータの真正性を検証するデータ検証装置とからなるデータ処理装置であって、

上記データ検証装置は

第 1 のデータを保持する基準値記憶部と、

第 2 のデータを保持する第 1 の副検査対象データ記憶部と、

上記第 1 の副検査対象データ記憶部に記憶されたデータから復号鍵を作成する復号鍵生成部と、

上記データ生成装置から送信されたデータを上記復号鍵生成部が生成した復号鍵で復号する復号部と、

上記復号部で復号されたデータが上記基準値記憶部に記憶されている上記第 1 のデータと所定の関係にあることを検査する検査部とからなり、

上記データ生成装置は、

上記データ検証装置から送付された上記第 1 のデータから第 3 のデータを作成する主検査対象データ生成部と、

第 4 のデータを保持する第 2 の副検査対象データ記憶部と、

上記第 2 の副検査対象データ記憶部に記憶されたデータから暗号鍵を作成する暗号鍵生成部と、

上記主検査対象データ生成部が作成した上記第 3 のデータを上記暗号鍵生成部

が生成した暗号鍵で暗号化する暗号部とからなり、

上記データ検証装置が、上記基準値記憶部に記憶されている上記第 1 のデータを上記データ生成装置に送信し、

上記データ生成装置が、上記主検査対象データ生成部で、上記データ検証装置から送信された上記第 1 のデータから上記第 3 のデータを生成し、さらに上記第 3 のデータを上記暗号部で暗号化して生成したデータを上記データ検証装置に送信し、

上記データ検証装置が、上記データ生成装置から送信されたデータを上記復号部で復号し、復号結果が上記基準値記憶部に記憶されている上記第 1 のデータと所定の関係にあることを上記検査部で検査することを特徴とするデータ処理装置

【請求項 2 0】 上記データ生成装置が生成する上記第 3 のデータが、上記データ検証装置から送信された上記第 1 のデータを所定の復号鍵で復号したものであり、上記データ検証装置の上記検証部が、上記データ生成装置から送信されたデータの復号結果が、上記第 1 のデータに対する復号結果であることを検査することを特徴とする請求項 1 9 記載のデータ処理装置。

【請求項 2 1】 上記データ生成装置が生成する上記第 3 のデータが、上記データ検証装置から送信された上記第 1 のデータに所定の署名鍵で署名して生成した署名値であり、上記データ検証装置の上記検証部が、上記データ生成装置から送信されたデータの復号結果が、上記基準値に対する正しい署名値であることを検査することを特徴とする請求項 1 9 記載のデータ処理装置。

【請求項 2 2】 上記データ生成装置が、乱数値を保持する第 1 のコミットメント乱数記憶部と、コミットメント乱数値からコミットメント値を生成するコミットメント生成部とを更に備え、

上記データ検証装置が、上記データ生成装置から送信されたコミットメント値を記憶するコミットメント情報記憶部を更に備え、

上記データ生成装置が、上記データ検証装置から上記第 1 のデータを受け取る前に、上記コミットメント生成部が生成したコミットメント値を上記データ検証装置に送信し、

主検査対象生成部において被検証値である上記第 3 のデータを生成する際には、さらに、上記コミットメント乱数記憶部に記憶された乱数値も使用し、

上記データ検証装置が、上記検査部での検査の際に、上記コミットメント情報記憶部に記憶された上記コミットメント値も使用することを特徴とする請求項 1 9 記載のデータ処理装置。

【請求項 2 3】 上記データ検証装置の上記復号鍵生成部が一方向性関数で構成されており、該一方向性関数に上記第 1 の副検査対象データ記憶部に記憶されたデータを入力して得られた結果が上記復号鍵であり、上記データ生成装置の上記暗号鍵生成部が上記データ検証装置の上記復号鍵生成部のものと同じ一方向性関数で構成されており、該一方向性関数に上記第 2 の副検査対象データ記憶部に記憶されたデータを入力して得られた結果が上記暗号鍵であることを特徴とする請求項 1 9 記載のデータ処理装置。

【請求項 2 4】 上記データ検証装置が、
前鍵を保持する第 1 の前鍵保持部を更に備え、
上記復号鍵生成部が上記復号鍵を生成する際に、上記第 1 の前鍵保持部に記憶された上記前鍵も使用し、
上記データ作成装置が、
前鍵を保持する第 2 の前鍵保持部を更に備え、
上記暗号鍵生成部が上記暗号鍵を生成する際に、上記第 2 の前鍵保持部に記憶された上記前鍵も使用することを特徴とする請求項 1 9 記載のデータ処理装置。

【請求項 2 5】 上記データ生成装置が、
上記前鍵を暗号化するための前鍵暗号鍵を保持する前鍵暗号鍵記憶部と、
上記前鍵暗号鍵記憶部に記憶された暗号鍵で上記前鍵を暗号化する前鍵暗号部とを備え、
上記データ検証装置が、
暗号化された上記前鍵を復号する前鍵復号鍵を記憶する前鍵復号鍵記憶部と、
上記前鍵復号鍵記憶部に記憶された上記前鍵復号鍵で暗号化された上記前鍵を復号する前鍵復号部とを備え、
上記データ生成装置が、上記前鍵を決定して、その前鍵を上記前鍵暗号鍵記憶

部に記憶された暗号鍵を使用して上記前鍵暗号部で暗号化し、その結果を上記データ検証装置に送信し、

上記データ検証装置が、上記データ生成装置から送信された暗号化された上記前鍵を上記前鍵復号鍵記憶部に記憶された上記前鍵復号鍵を使用して上記前鍵復号部で復号し、その結果を上記前鍵記憶部に記憶することを特徴とする請求項 2 4 記載のデータ処理装置。

【請求項 2 6】 上記データ検証装置が、上記第 1 の副検査対象データ記憶部に保持されているデータを上記データ生成装置に送信し、

上記データ生成装置は、上記データ検証装置から送信された該データを上記第 2 の副検査対象データ記憶部に記憶して、上記暗号鍵の生成に使用することを特徴とする請求項 1 9 記載のデータ処理装置。

【請求項 2 7】 上記データ生成装置が、上記第 2 の副検査対象データ記憶部に保持されているデータを上記データ検証装置に送信し、上記データ検証装置は、上記データ生成装置から送信された該データを上記第 1 の副検査対象データ記憶部に記憶して、上記復号鍵の生成に使用することを特徴とする請求項 1 9 記載のデータ処理装置。

【請求項 2 8】 上記暗号部で行われる暗号化が、上記暗号鍵を鍵とした共通鍵暗号の暗号処理であり、上記復号部で行われる復号が、上記復号鍵を鍵とした共通鍵暗号の復号処理であることを特徴とする請求項 1 9 記載のデータ処理装置。

【請求項 2 9】 上記暗号部で行われる暗号化が、所定の法数のもとでの第 3 のデータと上記暗号鍵の乗除算であり、上記復号部で行われる復号が、所定の法数のもとでの上記復号鍵と上記データ生成装置から送信されたデータの乗除算であることを特徴とする請求項 1 9 記載のデータ処理装置。

【請求項 3 0】 第 1 のデータ記憶手段と暗号手段とを具備する第 1 の装置と、第 2 のデータ記憶手段と復号手段と検証手段とを具備する第 2 の装置とを有し、上記第 1 の装置は、上記第 1 のデータ記憶手段に記憶されているデータに基づいて所定の検証対象データを上記暗号手段により暗号化し、上記第 2 の装置は上記暗号化された上記所定の検証対象データを上記第 2 のデータ記憶手段に記憶

されているデータに基づいて上記復号手段により復号し、復号結果の真正性を上記検証手段で検証し、検証に成功した場合には上記第 1 のデータ記憶手段に記憶されているデータと上記第 2 のデータ記憶手段に記憶されているデータの同一性を認証することを特徴とするデータ処理装置。

【請求項 3 1】 上記第 1 のデータ記憶手段に記憶されているデータの少なくとも一部は上記第 2 の装置から送られてきたデータとする請求項 3 0 記載のデータ処理装置。

【請求項 3 2】 上記第 2 のデータ記憶手段に記憶されているデータの少なくとも一部は上記第 1 の装置から送られてきたデータとする請求項 3 0 または 3 1 記載のデータ処理装置。

【発明の詳細な説明】

【0 0 0 1】

【発明の属する利用分野】

この発明はデータ作成およびデータ検証技術に関し、とくに、データの暗号化・復号を行ないながらデータの検証手続を行う際にデータのやり取りを行うパーティ間のデータの真正性または同一性を簡易に検証できるようにしたものである。

【0 0 0 2】

【従来の技術】

メッセージの真正性 (i n t e g r i t y) を保証するための暗号学的方法としては、メッセージ本体に、真正性チェック (i n t e g r i t y c h e c k) 用のデータを添付するものが知られている。添付されるデータはメッセージ認証コード (M e s s a g e A u t h e n t i c a t i o n C o d e : M A C) あるいはシール (s e a l) と呼ばれる (以降では MAC に統一する)。

【0 0 0 3】

MAC を使用した真正性チェックの方法を以下に示す。

- ①メッセージの作者と受信者はともに、秘密の鍵 k とシールを作成する際に使用するアルゴリズム $H ()$ を共有している。
- ②メッセージの作成者は、メッセージ m から MAC、 $\sigma = H (m, k)$ を作成し

、 m に σ を添付して受信者に送る。

③MAC σ が添付されたメッセージ m を受け取った受信者は、 $H(m, k)$ が σ と等しいかどうかを確認し、等しければメッセージの真正性が保たれていることがわかる。

【0004】

上記の方法は、たとえば、「Warwick Ford. Computer Communication Security, Prentice Hall 1, 1994, pp. 75-80」に開示されている。

【0005】

【発明が解決しようとする課題】

従来の技術では、MACをメッセージに添付しなければならないために、送信されるデータ量がメッセージより増加してしまう。通常 $H()$ にはMD5（メッセージダイジェスト5）やSHA-1（セキュアハッシュアルゴリズム）等の暗号学的ハッシュアルゴリズムが使用されるが、これによって作成されるMACは、MD5で16 byte、SHA-1で20 byteである。

【0006】

メッセージ自体が文書のように大きかったり、通信速度の大きい環境下で通信する場合には、この増加はさほど問題にはならないが、少量のデータを低速度の通信環境で送信する場合には、無視できなくなる。

【0007】

【課題を解決する手段】

ここで提案する発明は、メッセージが所定の特徴を持つデータを含んでいる場合に有効である。その特徴とは、そのデータが、受信者によってチェック可能な特定の条件を満たすことがわかっているというものである。

【0008】

たとえば、そのデータが、特定のパターンの繰り返しといった冗長度を持つ場合や、そのデータがデジタル署名の署名値のように、特定の検証式を満たすことがわかっている場合がこれにあたる。

【0009】

本発明のデータ生成装置は、所定の特徴を持つデータであって、その特徴を持つことが検証可能なデータである主検査対象データと、それ自身が特定の特徴を持つ必要がない（実際はなんらかの特徴を持つかもしれないが、別に持っていなくてもかまわない）副検査対象データを含むデータとを作成する。該データ生成装置は、データを暗号化する暗号部を持ち、主検査対象データを暗号部で暗号化する。暗号化のための鍵は、該データ生成装置が持つ暗号鍵生成部で、該データ生成装置内に保持されている副検査対象データから生成される。

【0010】

上記のデータ生成装置によって生成されたデータの真正性の検証を行うデータ検証装置は、該データ生成装置の暗号鍵生成部と同じ鍵生成アルゴリズムを持つ復号鍵生成部を持っており、さらに被検証値が所定の特徴を持つことを検証する検証部を持つ。

【0011】

このデータ検証装置は、データの検証の際に、副検査対象データから復号鍵を生成し、その鍵で暗号化された主検査対象データを復号し、復号結果が所定の特徴を持つことを検証する。

【0012】

データ作成装置が、副検査対象データの一部をデータ検証装置が所持していることを知っている場合には、データ作成装置が所持する副検査対象データのすべてをデータ検証装置に送付する必要はない。

【0013】

この検証に失敗した場合、以下のいずれかが起こっていることがデータ検証装置にわかる。

1. データ検証装置が持つ副検査対象データとデータ生成装置が持つ副検査対象データが異なっている。すなわち、副検査対象データの共有に失敗している。
2. たとえば、副検査対象データが検証者とデータ生成装置との間の送受信によって共有される場合には、送受信の過程でなんらかの差し替えが行われた可能性がある。
3. また、データ作成装置が、副検査対象データの一部をデータ検証装置が所持

していることを想定して、それを送付しなかった場合には、データ作成装置が想定しているのと同じデータをデータ検証装置が所持していない可能性がある。

4. データ生成装置自体が所定の特徴を持つ主検査対象データをつくることができない。たとえば、所定の特徴を持つ主検査対象データを作れるのが、特定の情報を持つものに限られている場合、データ生成装置はその範疇に該当しない。

【0014】

この方式によれば、データ生成装置が生成するデータには、真正性保証のためのMACに当たる情報は、暗号化された主検査対象データに畳み込まれており、MACをデータに添付する必要性をなくしている。

【0015】

本発明で用いる暗号鍵生成部あるいは復号鍵生成部は、一方向性関数で構成するのが一般的である。

【0016】

MD5あるいはSHA-1等のアルゴリズムが公開された一方向性ハッシュ関数を使用した場合、本発明のデータ生成装置が作成したデータについては、故意に改竄行為がなされた可能性を除いて、データに問題がないことが保証できる。

【0017】

もし、非公開の一方向性関数を使用した場合には、データがその非公開の一方向性関数を所持しているデータ作成装置で生成されたものであるかどうかを検査できる。すなわち、非公開の一方向性関数を共有している者以外による故意の改竄を排除できる。ただし、検査ができるのは、同じ非公開の一方向性関数を共有しているデータ検証装置のみである。

【0018】

また、この構成をとった場合の付加的な利点として、非公開の一方向性関数を共有しているもの以外には、主検査対象データを秘密にできるということがある。

【0019】

データ作成装置とデータ検証装置が秘密のデータ（前鍵と呼ぶ）を共有し、暗号鍵生成や復号鍵生成の際にその前鍵を使用することでも、同様の効果を得るこ

とができる。

【0020】

この方法でも、検査ができるのは、同じ前鍵を共有しているデータ検証装置のみであるが、データ作成装置からデータ検証装置に前鍵を送ることで、任意のデータ検証装置にデータの検査を行わせることが可能になる。当然、前鍵は暗号化されたデータ検証装置に送付されなければならないが、その暗号に公開鍵暗号をもちいれば、暗号鍵が公開されているどのデータ検証装置にもデータの検証を実行させることができる。

【0021】

本発明をさらに詳細に説明する。

【0022】

本発明によれば、データの真正性を低コストに検証するために、データ生成装置と、該生成装置で生成されたデータの真正性を検証するデータ検証装置とからなるデータ処理装置を構成する。そして、上記データ検証装置には、第1のデータを保持する基準値記憶部と、第2のデータを保持する第1の副検査対象データ記憶部と、上記第1の副検査対象データ記憶部に記憶されたデータから復号鍵を作成する復号鍵生成部と、上記データ生成装置から送信されたデータを上記復号鍵生成部が生成した復号鍵で復号する復号部と、上記復号部で復号されたデータが上記基準値記憶部に記憶されている上記第1のデータと所定の関係にあることを検査する検査部とを設ける。上記データ生成装置には、上記データ検証装置から送付された上記第1のデータから第3のデータを作成する主検査対象データ生成部と、第4のデータを保持する第2の副検査対象データ記憶部と、上記第2の副検査対象データ記憶部に記憶されたデータから暗号鍵を作成する暗号鍵生成部と、上記主検査対象データ生成部が作成した上記第3のデータを上記暗号鍵生成部が生成した暗号鍵で暗号化する暗号部とを設ける。上記データ検証装置が、上記基準値記憶部に記憶されている上記第1のデータを上記データ生成装置に送信し、上記データ生成装置が、上記主検査対象データ生成部で、上記データ検証装置から送信された上記第1のデータから上記第3のデータを生成し、さらに上記第3のデータを上記暗号部で暗号化して生成したデータを上記データ検証装置に

送信し、上記データ検証装置が、上記データ生成装置から送信されたデータを上記復号部で復号し、復号結果が上記基準値記憶部に記憶されている上記第 1 のデータと所定の関係にあることを上記検査部で検査する。

【 0 0 2 3 】

この構成においては、主検査対象データの検証を行いつつ、その検証手続に上記副検査対象データを畳み込んでいるので、副検査対象データの検証を別途行う必要がなく、また、MAC等を付加する必要もない。

【 0 0 2 4 】

なお、データ生成装置が生成するデータは、暗号部による暗号化結果および第 2 の副検査対象データ記憶部の記憶内容の双方を同時に含むデータでもよい。あるいは、データ生成装置が、暗号部による暗号化結果を含むデータと、第 2 の副検査対象データ記憶部の記憶内容を含むデータとを、個別のタイミングで生成して、これらが別々のタイミングでデータ検証装置に送られるようにしてもよい。

【 0 0 2 5 】

また、副検査対象データはどちらから送信するようにしてもよく、また、データ生成装置およびデータ検証装置の双方に第 3 者から送られたものでもよい。また、データのソースが一部はデータ生成装置、一部はデータ検証装置また一部はそれ以外であるような場合も適用できる。

【 0 0 2 6 】

また、この構成において、上記データ生成装置が生成する上記第 3 のデータが、上記データ検証装置から送信された上記第 1 のデータを所定の復号鍵で復号したものであり、上記データ検証装置の上記検証部が、上記データ生成装置から送信されたデータの復号結果が、上記第 1 のデータに対する復号結果であることを検査するようにしてもよい。

【 0 0 2 7 】

また、上記データ生成装置が生成する上記第 3 のデータが、上記データ検証装置から送信された上記第 1 のデータに所定の署名鍵で署名して生成した署名値であり、上記データ検証装置の上記検証部が、上記データ生成装置から送信されたデータの復号結果が、上記基準値に対する正しい署名値であることを検査するよ

うにしてもよい。

【0028】

また、上記データ生成装置が、乱数値を保持する第1のコミットメント乱数記憶部と、コミットメント乱数からコミットメント値を生成するコミットメント生成部とを更に備え、上記データ検証装置が、上記データ生成装置から送信されたコミットメント値を記憶するコミットメント情報記憶部を更に備え、上記データ生成装置が、上記データ検証装置から上記第1のデータを受け取る前に、上記コミットメント生成部が生成したコミットメント値を上記データ検証装置に送信し、主検査対象生成部において被検証値である上記第3のデータを生成する際には、さらに、上記コミットメント乱数記憶部に記憶された乱数値も使用し、上記データ検証装置が、上記検査部での検査の際に、上記コミットメント情報記憶部に記憶された上記コミットメント値も使用するようにしてもよい。

【0029】

また、上記データ検証装置の上記復号鍵生成部が一方向性関数で構成されており、該一方向性関数に上記第1の副検査対象データ記憶部に記憶されたデータを入力して得られた結果が上記復号鍵であり、上記データ生成装置の上記暗号鍵生成部が上記データ検証装置の上記復号鍵生成部のものと同じ一方向性関数で構成されており、該一方向性関数に上記第2の副検査対象データ記憶部に記憶されたデータを入力して得られた結果が上記暗号鍵であるようにしてもよい。

【0030】

また、上記データ検証装置が、前鍵を保持する第1の前鍵保持部を更に備え、上記復号鍵生成部が上記復号鍵を生成する際に、上記第1の前鍵保持部に記憶された上記前鍵も使用し、上記データ作成装置が、前鍵を保持する第2の前鍵保持部を更に備え、上記暗号鍵生成部が上記暗号鍵を生成する際に、上記第2の前鍵保持部に記憶された上記前鍵も使用するようにしてもよい。

【0031】

また、上記データ生成装置が、上記前鍵を暗号化するための前鍵暗号鍵を保持する前鍵暗号鍵記憶部と、上記前鍵暗号鍵記憶部に記憶された暗号鍵で上記前鍵を暗号化する前鍵暗号部とを備え、上記データ検証装置が、暗号化された上記前

鍵を復号する前鍵復号鍵を記憶する前鍵復号鍵記憶部と、上記前鍵復号鍵記憶部に記憶された上記前鍵復号鍵で暗号化された上記前鍵を復号する前鍵復号部とを備え、上記データ生成装置が、上記前鍵を決定して、その前鍵を上記前鍵暗号鍵記憶部に記憶された暗号鍵を使用して上記前鍵暗号部で暗号化し、その結果を上記データ検証装置に送信し、上記データ検証装置が、上記データ生成装置から送信された暗号化された上記前鍵を上記前鍵復号鍵記憶部に記憶された上記前鍵復号鍵を使用して上記前鍵復号部で復号し、その結果を上記前鍵記憶部に記憶するようにしてもよい。

【0032】

また、上記データ検証装置が、上記第1の副検査対象データ記憶部に保持されているデータを上記データ生成装置に送信し、上記データ生成装置は、上記データ検証装置から送信された該データを上記第2の副検査対象データ記憶部に記憶して、上記暗号鍵の生成に使用するようにしてもよい。

【0033】

また、上記データ生成装置が、上記第2の副検査対象データ記憶部に保持されているデータを上記データ検証装置に送信し、上記データ検証装置は、上記データ生成装置から送信された該データを上記第1の副検査対象データ記憶部に記憶して、上記復号鍵の生成に使用するようにしてもよい。

【0034】

また、上記暗号部で行われる暗号化が、上記暗号鍵を鍵とした共通鍵暗号の暗号処理であり、上記復号部で行われる復号が、上記復号鍵を鍵とした共通鍵暗号の復号処理であるようにしてもよい。

【0035】

また、上記暗号部で行われる暗号化が、所定の法数のもとでの第3のデータと上記暗号鍵の乗除算であり、上記復号部で行われる復号が、所定の法数のもとでの上記復号鍵と上記データ生成装置から送信されたデータの乗除算であるようにしてもよい。

【0036】

なお、本発明は、データ生成装置やデータ検証装置としても実現でき、またそ

れらの方法としても実現できる。また本発明の少なくとも一部をコンピュータプログラム製品として実現することもできる。

【0037】

【発明の実施の態様】

以下、この発明の実施例について説明する。

【0038】

〔第1の実施例〕

図1は、本発明を適用した第1の実施例の構成図である。

【0039】

本実施例は、データを生成するデータ生成器100と、データ生成器100で生成されたデータの正当性を検証するデータ検証器200からなるシステムである。データ生成器100とデータ検証器200の間ではデータの送受信が行われる。送受信はインターネット経由、イントラネット経由、電話回線経由、同一計算機内でのプロセス間通信等、種々の方法が可能である。基本的な機能は、データ検証器200が送付したデータに対してデータ生成器100が署名値を計算し、データ検証器200に送り返すものである。

【0040】

また、このデータ送受信の過程で、データ検証器200からデータ生成器100への情報の送付、およびその逆方向の情報の送付も行うことができる。具体的には、署名を依頼した日時情報をデータ検証器200からデータ生成器100に送付することができるし、署名を生成したデータ生成器100の識別子をデータ生成器100からデータ検証器200に送ることができる。

【0041】

この過程で、データ検証器200は、自分が送付した情報が改ざんされことなくデータ生成器100に届いたこと、情報を送付してきたデータ生成器100が署名を作成したデータ生成器100と同一であり、そのデータに改ざんが施されていないことを確認できる。

【0042】

データ生成器100が含むモジュールの役割について以下に記す。

〔受信部 1 0 1〕：データ検証器 2 0 0 から送付されたデータを受け取り、通信用フォーマットからデータ生成器 1 0 0 内部のフォーマットに変換した後、受信したデータをデータ生成器 1 0 0 内のほかのモジュールに振り分ける。

〔送信部 1 0 2〕：データ生成器 1 0 0 の各モジュールから受け取ったデータを通信に適したフォーマットに変換しデータ検証器 2 0 0 に送付する。

〔署名生成部 1 0 3〕：受信部 1 0 1 経由でデータ検証器 2 0 0 から送付された被署名値（署名の対象となるデータ）に対する署名値を作成するモジュールである。本実施例でデータ生成器 1 0 0 が作成する署名は、データ検証器 2 0 0 から送信された被署名値に S H A - 1 を適用した結果に R S A (R i v e s t - S h a m i r - A d l e m a n) 署名を施したものである。署名生成部 1 0 3 は、S H A - 1 のハッシュを実行するプログラムと、R S A の署名鍵を内蔵している。

〔暗号部 1 0 4〕：署名生成部 1 0 3 で生成された署名値を暗号化するモジュールである。このモジュールで行う暗号化は、データ検証部 2 0 0 の暗号文復号部 2 0 4 が復号に使用するのと同じ R S A 法数のもとの、鍵生成部 1 0 5 で生成される鍵と署名生成部 1 0 3 が生成した署名値との剰余演算である。

〔鍵生成部 1 0 5〕：暗号部 1 0 4 で使用される暗号鍵を生成するモジュールである。このモジュールは S H A - 1 等のハッシュ関数で構成されている。後述する前鍵記憶部 1 0 6 に記憶されている前鍵と、鍵生成用データ記憶部 1 0 9 に記憶されているデータに、このハッシュ関数を適用した結果が鍵となる。

〔前鍵生成部 1 0 6〕：鍵生成部 1 0 5 での鍵の生成に使用される前鍵を生成するモジュールである。生成する前鍵を攻撃者に予測されないよう、乱数生成器等を使って前鍵を生成する。

【 0 0 4 3 】

前鍵は、データ検証器 2 0 0 に送られ、データ生成器 1 0 0 とデータ検証器 2 0 0 の間で共有される。共有されている間は、データ検証器 2 0 0 に前鍵を送る必要はない。いつ共有をやめて、新しい前鍵をデータ検証器 2 0 0 に送るかは前鍵生成部 1 0 6 が決定する。そのため前鍵生成部 1 0 6 は、時計を内蔵しており、以前前鍵を送付してから一定時間が過ぎると、新たな前鍵を生成する。

〔前鍵暗号部 1 0 7〕：前鍵を暗号化し、その結果を送信部 1 0 2 に送るモジュ

ールである。このモジュールでの暗号化はRSA暗号を用いる。このモジュールはRSA暗号を実施するプログラムと、暗号鍵を内蔵している。

【前鍵記憶部 1 0 8】：前鍵生成部 1 0 6 が作成した前鍵を記憶するモジュールである。このモジュールが記憶している前鍵は、鍵生成部 1 0 5 が鍵を生成する際、および、前鍵暗号部 1 0 7 が前鍵を暗号化する際に参照される。

【鍵生成用データ記憶部 1 0 9】：鍵生成部 1 0 5 での鍵の生成に使用されるデータを記憶しているモジュールである。具体的には、署名依頼日時の情報と、このデータ生成器 1 0 0 の識別子を記憶している。前者のデータは、データ検証器 2 0 0 から送信されるものである。後者のデータは、本データ生成器 1 0 0 に埋め込まれたものであり、送信部 1 0 2 を通して、データ検証器 2 0 0 に送信される。

【 0 0 4 4 】

データ検証器 2 0 0 が含むモジュールの役割について以下に記す。

【送信部 2 0 1】：データを通信に適したフォーマットに変換しデータ生成器 1 0 0 に送付する。

【受信部 2 0 2】：データ生成器 1 0 0 から送付されたデータを受け取り、通信用フォーマットからデータ検証器 2 0 0 内部のフォーマットに変換した後、受信したデータをデータ検証器 2 0 0 内のほかのモジュールに振り分ける。

【被署名値記憶部 2 0 3】：このモジュールにデータ生成器 1 0 0 に署名を依頼するデータが記憶されている。ここに記憶されているデータは、送信部 2 0 1 を通してデータ生成器 1 0 0 に送られる。また、署名検証部 2 0 5 が署名値を検証する際に参照される。

【復号部 2 0 4】：データ生成器 1 0 0 から送信されたデータに含まれる暗号化された署名値を復号するモジュールである。ここで行われる復号は、署名検証部 2 0 5 での署名値の検証に使われるRSA法数のもとで、暗号化された署名値を鍵生成部 2 0 6 で生成される鍵で割る演算である。

【署名検証部 2 0 5】：復号部 2 0 4 で復号した署名値の正当性を検証するモジュールである。このモジュールは、SHA-1 を実行するプログラムと、データ生成器 1 0 0 の署名生成部 1 0 3 が所持する署名鍵に対応する検証鍵を内蔵して

いる。被署名値記憶部 2 0 3 に記憶されているデータに S H A - 1 を適用した結果が署名値を検証鍵でべき乗剰余した結果と同一かどうかで、署名値の正当性を検証する。

〔鍵生成部 2 0 6〕：復号部 2 0 4 が使用する復号鍵を作成するモジュールである。データ生成器 1 0 0 の鍵生成部 1 0 5 が持つのと同じハッシュ関数で構成されている。後述する前鍵記憶部 2 0 7 に記憶されている前鍵と、鍵生成用データ記憶部 2 0 9 に記憶されているデータに、このハッシュ関数を適用した結果が鍵となる。

〔前鍵記憶部 2 0 7〕：前鍵復号部 2 0 8 が復号した前鍵を記憶するモジュールである。このモジュールが記憶している前鍵は、鍵生成部 2 0 6 が鍵を生成する際に参照される。

〔前鍵復号部 2 0 8〕：前鍵は暗号化された状態でデータ生成器 1 0 0 から送信される。このモジュールは、暗号化された前鍵を復号する。このモジュールでの復号には R S A 復号を用いる。その復号鍵は、データ生成器 1 0 0 の前鍵暗号部 1 0 7 が保持する暗号鍵に対応する復号鍵であり、本モジュールはこの復号鍵を内蔵している。

〔鍵生成用データ記憶部 2 0 9〕：鍵生成部 2 0 6 での鍵の生成に使用されるデータを記憶しているモジュールである。具体的には、署名依頼日時の情報と、署名を生成したデータ生成器 1 0 0 の識別子を記憶している。前者のデータは、本データ検証器 2 0 0 に付随したクロックから取り出される。後者のデータは、データ生成器 1 0 0 から署名値とともに送信される。

【 0 0 4 5 】

本実施例の動作について説明する。

【 0 0 4 6 】

本実施例は、大まかには、データ検証器 2 0 0 がデータ生成器 1 0 0 に被署名値を含むデータ（署名作成依頼と呼ぶ）を送付し、データ生成器 1 0 0 がデータ検証器 2 0 0 に署名値を含むデータ（署名データと呼ぶ）を返信し、データ検証器 2 0 0 で署名データの正当性を検査するというステップを踏む。

【 0 0 4 7 】

データ検証器 2 0 0 が署名作成依頼を作成するときの動作を以下に示す。なお、この手順自体は簡潔なのでとくに図示しない。

〔前提〕：被署名データが被署名値記憶部 2 0 3 に記憶されている状態からスタートする。

〔ステップ 1〕：データ検証器 2 0 0 が内蔵している時計から現在時刻情報を取り出し、それを署名依頼時刻として鍵生成用データ記憶部 1 0 9 に記憶する。

〔ステップ 2〕：被署名値記憶部 2 0 3 に記憶されている被署名値と、鍵生成用データ記憶部 2 0 9 に記憶されている署名依頼時刻とから署名作成依頼を作成し、送信部 2 0 1 からデータ生成器 1 0 0 に送付する。

【 0 0 4 8 】

署名作成依頼を受け取ったデータ生成器 1 0 0 の動作を以下に示す。また、この動作の手順を記述したフローチャートを図 2 に示す。

〔ステップ S 1 0 1〕：受信部 1 0 1 で受信した署名作成依頼から、署名依頼時刻を取り出し、鍵生成用データ記憶部 1 0 9 に記憶する。

〔ステップ S 1 0 2〕：受信部 1 0 1 で受信した署名作成依頼から、被署名値を取り出して署名生成部 1 0 3 に送付し、署名生成部 1 0 3 で、被署名値に対する署名値を作成する。

〔ステップ S 1 0 3〕：もし、新しい前鍵の生成が必要なら、前鍵生成部 1 0 6 で前鍵を作成し、前鍵記憶部 1 0 8 に記憶する。必要でなければ、ステップ S 1 0 5 へ進む。

〔ステップ S 1 0 4〕：前鍵暗号部 1 0 7 で前鍵記憶部 1 0 8 に記憶されている前鍵を暗号化する。

〔ステップ S 1 0 5〕：鍵生成部 1 0 5 で、前鍵記憶部 1 0 8 に記憶されている前鍵と、鍵生成用データ記憶部 1 0 9 に記憶されている署名依頼日時と本データ生成器 1 0 0 の識別子から、鍵を生成する。

〔ステップ S 1 0 6〕：署名生成部 1 0 3 で生成された署名値を、鍵生成部 1 0 5 で生成された鍵を用い、暗号部 1 0 4 で暗号化する。

〔ステップ S 1 0 7〕：暗号部 1 0 4 で暗号化された署名値と、もしあれば前鍵暗号部 1 0 7 で暗号化された前鍵と、鍵生成用データ記憶部 1 0 9 に記憶されて

いる本データ生成器 1 0 0 の識別子とから署名データを作成し、送信部 1 0 2 からデータ検証器 2 0 0 に送付する。

【 0 0 4 9 】

署名データを受け取ったデータ検証器 2 0 0 の動作を以下に示す。また、この動作の手順を記述したフローチャートを図 3 に示す。

【 0 0 5 0 】

〔ステップ S 1 1 1〕：受信部 2 0 2 で受信した署名データから、署名を生成したデータ生成器 1 0 0 の識別子を取り出し、鍵生成用データ記憶部 2 0 9 に記憶する。

〔ステップ S 1 1 2〕：署名データに暗号化された前鍵が含まれていれば、それを取り出し、前鍵復号部 2 0 8 で復号し、その結果を前鍵記憶部 2 0 7 に記憶する。もし、暗号化された前鍵が含まれていなければ、何もせずステップ S 1 1 3 へ進む。

〔ステップ S 1 1 3〕：鍵生成部 2 0 6 で、前鍵記憶部 2 0 7 に記憶されている前鍵と、鍵生成用データ記憶部 2 0 9 に記憶されている署名依頼日時とデータ生成器 1 0 0 の識別子から、鍵を生成する。

〔ステップ S 1 1 4〕：署名データから暗号化された署名値を取り出し、鍵生成部 2 0 6 で生成された鍵を使用して、復号部 2 0 4 で復号する。

〔ステップ S 1 1 5〕：復号部 2 0 4 での復号の結果が、被署名値記憶部 2 0 3 に記憶されている被署名値の正しい署名であるかどうかを署名検証部 2 0 5 で検査する。

【 0 0 5 1 】

データ検証器 2 0 0 が、ステップ S 1 1 5 の検査で、正しい署名値であることが確認できた場合、データ検証器 2 0 0 にとっては、以下のことが保証される。

【 0 0 5 2 】

1. データ生成器 1 0 0 が作成した署名値が、被署名値に対して正しい署名であること。
2. データ検証器 2 0 0 が送付した署名依頼日時が、差し替えられることなくデータ生成器 1 0 0 に届いたこと。

3. 受け取ったデータ生成器 1 0 0 の識別子が、署名を生成したデータ生成器 1 0 0 と同一のものであること。

【0 0 5 3】

[第 2 の実施例]

図 4 は、本発明を適用した第 2 の実施例の構成図である。

【0 0 5 4】

本実施例は、第 1 の実施例と同様に、データを生成するデータ生成器 1 0 0 と、データ生成器 1 0 0 で生成されたデータの正当性を検証するデータ検証器 2 0 0 からなるシステムである。データ生成器 1 0 0 とデータ検証器 2 0 0 の間ではデータの送受信が行われる。送受信はインターネット経由、イントラネット経由、電話回線経由、同一計算機内でのプロセス間通信等、種々の方法が可能である。

【0 0 5 5】

基本的な機能は、データ検証器 2 0 0 が送付した暗号文をデータ生成器 1 0 0 が復号し、その結果をデータ検証器 2 0 0 に送り返すものである。

【0 0 5 6】

また、このデータ送受信の過程で、データ検証器 2 0 0 からデータ生成器 1 0 0 への情報の送付、およびその逆方向の情報の送付も行うことができる。具体的には、復号を依頼した日時情報をデータ検証器 2 0 0 からデータ生成器 1 0 0 に送付することができるし、復号を実行したデータ生成器 1 0 0 の識別子をデータ生成器 1 0 0 からデータ検証器 2 0 0 に送ることができる。

【0 0 5 7】

この過程で、データ検証器 2 0 0 は、自分が送付した情報が改ざんされることなくデータ生成器 1 0 0 に届いたこと、情報を送付してきたデータ生成器 1 0 0 が復号を行ったデータ生成器 1 0 0 と同一であり、そのデータに改ざんが施されていないことを確認できる。

【0 0 5 8】

データ生成器 1 0 0 が含むモジュールの役割について以下に記す。

[受信部 1 0 1] : データ検証器 2 0 0 から送付されたデータを受け取り、通信

用フォーマットからデータ生成器 100 内部のフォーマットに変換した後、受信したデータをデータ生成器 100 内のほかのモジュールに振り分ける。

〔送信部 102〕：データ生成器 100 の各モジュールから受け取ったデータを通信に適したフォーマットに変換しデータ検証器 200 に送付する。

〔暗号文復号部 110〕：受信部 101 経由でデータ検証器 200 から送付された暗号文を復号するモジュールである。本実施例でデータ生成器 100 が採用する復号アルゴリズムは RSA である。このモジュールは、RSA の復号鍵を内蔵している。

〔暗号部 104〕：暗号文復号部 110 で生成された復号結果を暗号化するモジュールである。このモジュールで行われる暗号は DES-CBC (Data Encryption Standard-Cipher Block Chaining mode) である。このモジュールは DES-CBC を実行できるプログラムを内蔵している。

〔鍵生成部 105〕：暗号部 104 で使用される暗号鍵を生成するモジュールである。ここで生成される暗号鍵は、暗号部 104 で使用される DES-CBC の初期ベクタ 64 bit と暗号鍵 56 bit のあわせて 120 bit のデータである。このデータは、後述する前鍵記憶部 108 に記憶されている前鍵と、鍵生成用データ記憶部 109 に記憶されているデータに 120 bit のハッシュ値を出力する一方向性関数を施した結果である。したがって、このモジュールは 120 bit のハッシュ値を出力する一方向性関数を実行するプログラムを内蔵している。

〔前鍵生成部 106〕：鍵生成部 105 での鍵の生成に使用される前鍵を生成するモジュールである。生成する前鍵を攻撃者に予測されないよう、乱数生成器等を使って前鍵を生成する。

【0059】

前鍵は、データ検証器 200 に送られ、データ生成器 100 とデータ検証器 200 の間で共有される。共有されている間は、データ検証器 200 に前鍵を送る必要はない。いつ共有をやめて、新しい前鍵をデータ検証器 200 に送るかは前鍵生成部 106 が決定する。そのため前鍵生成部 106 は、時計を内蔵しており

、以前前鍵を送付してから一定時間が過ぎると、新たな前鍵を生成する。

〔前鍵暗号部 1 0 7〕：前鍵を暗号化し、その結果を送信部 1 0 2 に送るモジュールである。このモジュールでの暗号化は R S A 暗号を用いる。このモジュールは R S A 暗号を実施するプログラムと、暗号鍵を内蔵している。

〔前鍵記憶部 1 0 8〕：前鍵生成部 1 0 6 が作成した前鍵を記憶するモジュールである。このモジュールが記憶している前鍵は、鍵生成部 1 0 5 が鍵を生成する際、および、前鍵暗号部 1 0 7 が前鍵を暗号化する際に参照される。

〔鍵生成用データ記憶部 1 0 9〕：鍵生成部 1 0 5 での鍵の生成に使用されるデータを記憶しているモジュールである。具体的には、署名依頼日時の情報と、このデータ生成器 1 0 0 の識別子を記憶している。前者のデータは、データ検証器 2 0 0 から送信されるものである。後者のデータは、本データ生成器 1 0 0 に埋め込まれたものであり、送信部 1 0 2 を通して、データ検証器 2 0 0 に送信される。

【 0 0 6 0 】

データ検証器 2 0 0 が含むモジュールの役割について以下に記す。

〔送信部 2 0 1〕：データを通信に適したフォーマットに変換しデータ生成器 1 0 0 に送付する。

〔受信部 2 0 2〕：データ生成器 1 0 0 から送付されたデータを受け取り、通信用フォーマットからデータ検証器 2 0 0 内部のフォーマットに変換した後、受信したデータをデータ検証器 2 0 0 内のほかのモジュールに振り分ける。

〔暗号文記憶部 2 1 0〕：このモジュールにデータ生成器 1 0 0 に復号を依頼する暗号文が記憶されている。ここに記憶されているデータは、送信部 2 0 1 を通してデータ生成器 1 0 0 に送られる。ここに蓄積されているデータは、特定のフォーマットを持つ平文を R S A で暗号化したものであり、その暗号鍵は、データ生成器 1 0 0 の暗号文復号部 1 1 0 が内蔵する復号鍵に対応するものである。

〔復号部 2 0 4〕：データ生成器 1 0 0 から送信されたデータに含まれる暗号化されたデータを復号するモジュールである。ここで行われる復号は、D E S - C B C であり、このモジュールは D E S - C B C を実行するプログラムを内蔵している。

〔復号結果検証部 211〕：復号部 204 で復号したデータが、データ生成器 100 に復号を依頼した暗号文の復号結果であることを検査するモジュールである。データが所定のフォーマットをしていれば、それを正当なものであると判定する。

〔鍵生成部 206〕：復号部 204 が使用する復号鍵を作成するモジュールである。ここで生成される鍵は、復号部 204 で使用される DES-CBC の初期ベクタ 64 bit と復号鍵 56 bit のあわせて 120 bit のデータである。このデータは、後述する前鍵記憶部 207 に記憶されている前鍵と、鍵生成用データ記憶部 209 に記憶されているデータに、データ生成器 100 の鍵生成部 105 が持つと同じ 120 bit のハッシュ値を出力する一方向性関数を施した結果である。したがって、このモジュールは 120 bit のハッシュ値を出力するデータ生成器 100 の鍵生成部 105 が持つと同じ一方向性関数を実行するプログラムを内蔵している。

〔前鍵記憶部 207〕：前鍵復号部 208 が復号した前鍵を記憶するモジュールである。このモジュールが記憶している前鍵は、鍵生成部 206 が鍵を生成する際に参照される。

〔前鍵復号部 208〕：前鍵は暗号化された状態でデータ生成器 100 から送信される。このモジュールは、暗号化された前鍵を復号する。このモジュールでの復号には RSA 復号を用いる。その復号鍵は、データ生成器 100 の前鍵暗号部 107 が保持する暗号鍵に対応する復号鍵であり、本モジュールはこの復号鍵を内蔵している。

〔鍵生成用データ記憶部 209〕：鍵生成部 206 での鍵の生成に使用されるデータを記憶しているモジュールである。具体的には、署名依頼日時の情報と、署名を生成したデータ生成器 100 の識別子を記憶している。前者のデータは、本データ検証器 200 に付随したクロックから取り出される。後者のデータは、データ生成器 100 から暗号文とともに送信される。

【0061】

本実施例の動作について説明する。

【0062】

本実施例は、大まかには、データ検証器 2 0 0 がデータ生成器 1 0 0 に暗号文を含むデータ（復号依頼と呼ぶ）を送付し、データ生成器 1 0 0 がデータ検証器 2 0 0 に復号結果を含むデータ（復号データと呼ぶ）を返信し、データ検証器 2 0 0 で復号データの正当性を検査するというステップを踏む。

【0 0 6 3】

データ検証器 2 0 0 が復号依頼を作成するときの動作を以下に示す。なお、復号依頼の手順自体は簡潔であるので図示しない。

【前提】：暗号文が暗号文記憶部 2 1 0 に記憶されている状態からスタートする。

【ステップ 1】：データ検証器 2 0 0 が内蔵している時計から現在時刻情報を取り出し、それを復号依頼時刻として鍵生成用データ記憶部 2 0 9 に記憶する。

【ステップ 2】：暗号文記憶部 2 1 0 に記憶されている暗号文と、鍵生成用データ記憶部 2 0 9 に記憶されている復号依頼時刻とから復号依頼を作成し、送信部 2 0 1 からデータ生成器 1 0 0 に送付する。

【0 0 6 4】

復号依頼を受け取ったデータ生成器 1 0 0 の動作を以下に示す。また、この動作の手順を記述したフローチャートを図 5 に示す。

【0 0 6 5】

【ステップ S 1 2 1】：受信部 1 0 1 で受信した復号依頼から、復号依頼時刻を取り出し、鍵生成用データ記憶部 1 0 9 に記憶する。

【ステップ S 1 2 2】：受信部 1 0 1 で受信した復号依頼から、暗号文を取り出して暗号文復号部 1 1 0 に送付し、暗号文復号部 1 1 0 で、暗号文を復号する。

【ステップ S 1 2 3】：もし、新しい前鍵の生成が必要なら、前鍵生成部 1 0 6 で前鍵を作成し、前鍵記憶部 1 0 8 に記憶する。必要でなければ、ステップ S 1 2 5 へ進む。

【ステップ S 1 2 4】：前鍵暗号部 1 0 7 で前鍵記憶部 1 0 8 に記憶されている前鍵を暗号化する。

【ステップ S 1 2 5】：鍵生成部 1 0 5 で、前鍵記憶部 1 0 8 に記憶されている前鍵と、鍵生成用データ記憶部 1 0 9 に記憶されている復号依頼日時と本データ

生成器 1 0 0 の識別子から、鍵を生成する。

〔ステップ S 1 2 6〕：暗号文復号部 1 1 0 での復号結果を、鍵生成部 1 0 5 で生成された鍵を用いて、暗号部 1 0 4 で暗号化する。

〔ステップ S 1 2 7〕：暗号部 1 0 4 で暗号化された復号結果と、もしあれば前鍵暗号部 1 0 7 で暗号化された前鍵と、鍵生成用データ記憶部 1 0 9 に記憶されている本データ生成器 1 0 0 の識別子とから復号データを作成し、送信部 1 0 2 からデータ検証器 2 0 0 に送付する。

【 0 0 6 6 】

復号データを受け取ったデータ検証器 2 0 0 の動作を以下に示す。また、この動作の手順を記述したフローチャートを図 6 に示す。

【 0 0 6 7 】

〔ステップ S 1 3 1〕：受信部 2 0 2 で受信した復号データから、復号を実行したデータ生成器 1 0 0 の識別子を取り出し、鍵生成用データ記憶部 2 0 9 に記憶する。

〔ステップ S 1 3 2〕：復号データに暗号化された前鍵が含まれていれば、それを取り出し、前鍵復号部 2 0 8 で復号し、その結果を前鍵記憶部 2 0 7 に記憶する。もし、暗号化された前鍵が含まれていなければ、何もせずステップ S 1 3 3 へ進む。

〔ステップ S 1 3 3〕：鍵生成部 2 0 6 で、前鍵記憶部 2 0 7 に記憶されている前鍵と、鍵生成用データ記憶部 2 0 9 に記憶されている復号依頼日時とデータ生成器の識別子から、鍵を生成する。

〔ステップ S 1 3 4〕：復号データから暗号化された復号結果を取り出し、鍵生成部 2 0 6 で生成された鍵を使用して、復号部 2 0 4 で復号する。

〔ステップ S 1 3 5〕：復号部 2 0 4 での復号の結果が、所定のフォーマットのデータであるかどうかを復号結果検証部 2 1 1 で検査する。

【 0 0 6 8 】

データ検証器 2 0 0 が、ステップ S 1 3 5 の検査で、復号結果が所定のフォーマットを満たしていることが確認できた場合、データ検証器 2 0 0 にとっては、以下のことが保証される。

【0069】

1. データ生成器 100 が作成した復号結果が正しいこと。
2. データ検証器 200 が送付した復号依頼日時が、差し替えられることなくデータ生成器 100 に届いたこと。
3. 受け取ったデータ生成器 100 の識別子が、復号を実行したデータ生成器 100 と同一のものであること。

【0070】

なお、本実施例における所定のフォーマットまたはパターンとはどのようなものでもよい。実際の文書のパターンをも利用できる。例えば、文書のうち規則性のある部分を主検査対象データとし、その他の部分を副検査対象データとする。図 10 に示すような HTML (H y p e r t e x t M a r k u p L a n g u a g e) 文書においてはヘッダ部分を主検査対象データとし、ボディ部分を副検査対象データとして、主検査対象データのヘッダ部分の特徴を検査することにより、データ生成器およびデータ検証器のデータの同一性を検証できる。この場合、ヘッダ部分またはそれを含む部分を暗号化してデータ生成器に送り、その他の部分を、そのまま共有したいデータとして送り、あるいは送られることにより、効率のよいデータの共有化を保証できる。

【0071】

また、HTML 文書その他、他の SGML (S t a n d a r d G e n e r a l i z e d M a r k u p L a n g u a g e) 文書、例えば XML (E x t e n s i b l e M a r k u p L a n g u a g e) やそのスタイルシートにも適用できる。

【0072】

また、主検査対象データの特徴としては文字列自体を採用してもよい。すなわち、主検査対象データに含まれる所定の文字列があるかどうかを検証してもよい。

【0073】

[第 3 の実施例]

図 7 は、本発明を適用した第 3 の実施例の構成図である。

【0074】

本実施例は、データを生成するデータ生成器100と、データ生成器100で生成されたデータの正当性を検証するデータ検証器200からなるシステムである。データ生成器100とデータ検証器200の間ではデータの送受信が行われる。送受信はインターネット経由、イントラネット経由、電話回線経由、同一計算機内でのプロセス間通信等、種々の方法が可能である。

【0075】

基本的な機能は、データ検証器200がデータ生成器100を認証することであり、データ生成器100がコミットメントを作成してデータ検証器200に送付し、データ検証器200がチャレンジをデータ生成器100に送付し、チャレンジに対するレスポンスをデータ生成器100が計算し、データ検証器200に送り返すものである。

【0076】

また、このデータ送受信の過程で、データ検証器200からデータ生成器100への情報の送付、およびその逆方向の情報の送付も行うことができる。具体的には、認証を実行した日時情報をデータ検証器200からデータ生成器100に送付することができるし、レスポンスを生成したデータ生成器100の識別子をデータ生成器100からデータ検証器200に送ることができる。

【0077】

この過程で、データ検証器200は、自分が送付した情報が改ざんされことなくデータ生成器100に届いたこと、情報を送付してきたデータ生成器100が証明を行ったデータ生成器100と同一であり、そのデータに改ざんが施されていないことを確認できる。

【0078】

データ生成器100が含むモジュールの役割について以下に記す。

【受信部101】：データ検証器100から送付されたデータを受け取り、通信用フォーマットからデータ生成器100内部のフォーマットに変換した後、受信したデータをデータ生成器100内のほかのモジュールに振り分ける。

【送信部102】：データ生成器100の各モジュールから受け取ったデータを

通信に適したフォーマットに変換しデータ検証器 2 0 0 に送付する。

〔乱数生成部 1 2 0〕：コミットメントのもととなる乱数を生成するモジュールである。

〔乱数記憶部 1 2 1〕：乱数生成部 1 2 0 で生成された乱数を記憶するモジュールである。

〔コミットメント生成部 1 2 2〕：乱数記憶部 1 2 1 に記憶されている乱数をもとに、コミットメントを生成するモジュールである。

〔レスポンス生成部 1 2 3〕：受信部 1 0 1 経由でデータ検証器 2 0 0 から送付されたチャレンジに対するレスポンスを作成するモジュールである。本実施例でデータ生成器は Guil l o u - Q u i s q u a t e r 認証方式に基づいてレスポンスを生成する。レスポンス生成の際には、チャレンジだけでなく、乱数記憶部 1 2 1 に記憶されている乱数も使用される。

〔暗号部 1 0 4〕：レスポンス生成部 1 2 3 で生成されたレスポンスを暗号化するモジュールである。このモジュールで行う暗号化は、Guil l o u - Q u i s q u a t e r 認証で使用する法数のもとの、鍵生成部 1 0 5 で生成される鍵とレスポンス生成部 1 2 3 が生成したレスポンスとの剰余演算である。

〔鍵生成部 1 0 5〕：暗号部 1 0 4 で使用される暗号鍵を生成するモジュールである。このモジュールは S H A - 1 等のハッシュ関数で構成されている。後述する前鍵記憶部 1 0 8 に記憶されている前鍵と、鍵生成用データ記憶部 1 0 9 に記憶されているデータに、このハッシュ関数を適用した結果が鍵となる。

〔前鍵生成部 1 0 6〕：鍵生成部 1 0 5 での鍵の生成に使用される前鍵を生成するモジュールである。生成する前鍵を攻撃者に予測されないよう、乱数生成器等を使って前鍵を生成する。

【 0 0 7 9 】

前鍵は、データ検証器 2 0 0 に送られ、データ生成器 1 0 0 とデータ検証器 2 0 0 の間で共有される。共有されている間は、データ検証器 2 0 0 に前鍵を送る必要はない。いつ共有をやめて、新しい前鍵をデータ検証器 2 0 0 に送るかは前鍵生成部 1 0 6 が決定する。そのため前鍵生成部 1 0 6 は、時計を内蔵しており、以前前鍵を送付してから一定時間が過ぎると、新たな前鍵を生成する。

〔前鍵暗号部 1 0 7〕：前鍵を暗号化し、その結果を送信部 1 0 2 に送るモジュールである。このモジュールでの暗号化は R S A 暗号を用いる。このモジュールは R S A 暗号を実施するプログラムと、暗号鍵を内蔵している。

〔前鍵記憶部 1 0 8〕：前鍵生成部 1 0 6 が作成した前鍵を記憶するモジュールである。このモジュールが記憶している前鍵は、鍵生成部 1 0 5 が鍵を生成する際、および、前鍵暗号部 1 0 7 が前鍵を暗号化する際に参照される。

〔鍵生成用データ記憶部 1 0 9〕：鍵生成部 1 0 5 での鍵の生成に使用されるデータを記憶しているモジュールである。具体的には、認証実行日時の情報と、このデータ生成器 1 0 0 の識別子を記憶している。前者のデータは、データ検証器 2 0 0 から送信されるものである。後者のデータは、本データ生成器 1 0 0 に埋め込まれたものであり、送信部 1 0 2 を通して、データ検証器 2 0 0 に送信される。

【 0 0 8 0 】

データ検証器 2 0 0 が含むモジュールの役割について以下に記す。

〔送信部 2 0 1〕：データを通信に適したフォーマットに変換しデータ生成器 1 0 0 に送付する。

〔受信部 2 0 2〕：データ生成器 1 0 0 から送付されたデータを受け取り、通信用フォーマットからデータ検証器 2 0 0 内部のフォーマットに変換した後、受信したデータをデータ検証器 2 0 0 内のほかのモジュールに振り分ける。

〔コミットメント記憶部 2 2 0〕：このモジュールにデータ生成器 1 0 0 から送られたコミットメントを記憶する。ここに記憶されているデータは、レスポンス検証部 2 2 1 がレスポンスを検証する際に参照される。

〔チャレンジ記憶部 2 2 2〕：このモジュールにデータ生成器 1 0 0 にレスポンスを要求するチャレンジが記憶されている。ここに記憶されているデータは、送信部 2 0 1 を通してデータ生成器 1 0 0 に送られる。また、レスポンス検証部 2 2 1 がレスポンスを検証する際に参照される。

〔復号部 2 0 4〕：データ生成器 1 0 0 から送信されたデータに含まれる暗号化されたレスポンスを復号するモジュールである。ここで行われる復号は、レスポンス検証部 2 2 1 でのレスポンスの検証に使われる法数のもとで、暗号化された

レスポンスを鍵生成部 2 0 6 で生成される鍵で割る演算である。

〔レスポンス検証部 2 2 1〕：復号部 2 0 4 で復号したレスポンスの正当性を検証するモジュールである。このモジュールでは、コミットメント記憶部 2 2 0 に記憶されているコミットメント、チャレンジ記憶部 2 2 2 に記憶されているチャレンジに対して、復号部 2 0 4 で復号されたレスポンスが正当なものであるかどうかを検査する。

〔鍵生成部 2 0 6〕：復号部 2 0 4 が使用する復号鍵を作成するモジュールである。データ生成器 1 0 0 の鍵生成部 1 0 5 が持つのと同じハッシュ関数で構成されている。後述する前鍵記憶部 2 0 7 に記憶されている前鍵と、鍵生成用データ記憶部 2 0 9 に記憶されているデータに、このハッシュ関数を適用した結果が鍵となる。

〔前鍵記憶部 2 0 7〕：前鍵復号部 2 0 8 が復号した前鍵を記憶するモジュールである。このモジュールが記憶している前鍵は、鍵生成部 2 0 6 が鍵を生成する際に参照される。

〔前鍵復号部 2 0 8〕：前鍵は暗号化された状態でデータ生成器 1 0 0 から送信される。このモジュールは、暗号化された前鍵を復号する。このモジュールでの復号には R S A 復号を用いる。その復号鍵は、データ生成器 1 0 0 の前鍵暗号部 1 0 7 が保持する暗号鍵に対応する復号鍵であり、本モジュールはこの復号鍵を内蔵している。

〔鍵生成用データ記憶部 2 0 9〕：鍵生成部 2 0 6 での鍵の生成に使用されるデータを記憶しているモジュールである。具体的には、認証実行日時の情報と、署名を生成したデータ生成器 1 0 0 の識別子を記憶している。前者のデータは、本データ検証器 2 0 0 に付随したクロックから取り出される。後者のデータは、データ生成器 1 0 0 から証明データとともに送信される。

【0 0 8 1】

本実施例の動作について説明する。

【0 0 8 2】

本実施例は、大まかには、データ生成器 1 0 0 がコミットメントをデータ検証器 2 0 0 に送信し、データ検証器 2 0 0 がデータ生成器 1 0 0 にチャレンジを含

むデータ（チャレンジデータと呼ぶ）を送付し、データ生成器 1 0 0 がデータ検証器 2 0 0 にレスポンスを含むデータ（レスポンスデータと呼ぶ）を返信し、データ検証器 2 0 0 でレスポンスデータの正当性を検査するというステップを踏む。

【0 0 8 3】

データ生成器 1 0 0 がコミットメントを作成するときの動作を以下に示す。このコミットメント作成動作は簡潔なのでとくに図示することはない。

【ステップ 1】：乱数生成部 1 2 0 でコミットメント用の乱数を生成し、乱数記憶部 1 2 1 に記憶する。

【ステップ 2】：乱数記憶部 1 2 1 に記憶されている乱数からコミットメント生成部 1 2 2 でコミットメントを作成する。

【ステップ 3】：生成したコミットメントを送信部 1 0 2 からデータ検証器 2 0 0 に送付する。

【0 0 8 4】

コミットメントを受け取ったデータ検証器 2 0 0 の動作を以下に示す。このときのデータ検証器 2 0 0 の動作も簡潔なのでとくに図示することはない。

【ステップ 1】：受信部 2 0 1 で受信したコミットメントをコミットメント記憶部 2 2 0 に記憶する。

【ステップ 2】：チャレンジを生成し、チャレンジ記憶部 2 2 2 にチャレンジを記憶する。

【ステップ 3】：データ検証器 2 0 0 が内蔵している時計から現在時刻情報を取り出し、それを認証実行時刻として鍵生成用データ記憶部 2 0 9 に記憶する。

【ステップ 4】：チャレンジ記憶部 2 2 2 に記憶されているチャレンジと、鍵生成用データ記憶部 2 0 9 に記憶されている認証実行時刻とからチャレンジデータを作成し、送信部 2 0 1 からデータ生成器 1 0 0 に送付する。

【0 0 8 5】

チャレンジデータを受け取ったデータ生成器 1 0 0 の動作を以下に示すに示す。また、この動作の手順を記述したフローチャートを図 8 に示す。

【ステップ S 1 4 1】：受信部 1 0 1 で受信したチャレンジデータから、認証実

行時刻を取り出し、鍵生成用データ記憶部 1 0 9 に記憶する。

【ステップ S 1 4 2】：受信部 1 0 1 で受信したチャレンジデータから、チャレンジを取り出してレスポンス生成部 1 2 3 に送付し、レスポンス生成部 1 2 3 で、チャレンジと乱数記憶部 1 2 1 に記憶されている乱数とからレスポンスを作成する。

【ステップ S 1 4 3】：もし、新しい前鍵の生成が必要なら、前鍵生成部で前鍵 1 0 6 を作成し、前鍵記憶部 1 0 8 に記憶する。必要でなければ、ステップ S 1 4 5 へ進む。

【ステップ S 1 4 4】：前鍵暗号部 1 0 7 で前鍵記憶部 1 0 8 に記憶されている前鍵を暗号化する。

【ステップ S 1 4 5】：鍵生成部 1 0 5 で、前鍵記憶部 1 0 8 に記憶されている前鍵と、鍵生成用データ記憶部 1 0 9 に記憶されている認証実行日時と本データ生成器 1 0 0 の識別子から、鍵を生成する。

【ステップ S 1 4 6】：レスポンス生成部 1 2 3 で生成されたレスポンスを、鍵生成部 1 0 5 で生成された鍵を用いて、暗号部 1 0 4 で暗号化する。

【ステップ S 7】：暗号部 1 0 4 で暗号化されたレスポンスと、もしあれば前鍵暗号部 1 0 7 で暗号化された前鍵と、鍵生成用データ記憶部 1 0 9 に記憶されている本データ生成器 1 0 0 の識別子とからレスポンスデータを作成し、送信部 1 0 2 からデータ検証器 2 0 0 に送付する。

【0 0 8 6】

レスポンスデータを受け取ったデータ検証器 2 0 0 の動作を以下に示す。また、この動作の手順を記述したフローチャートを図 9 に示す。

【ステップ S 1 5 1】：受信部 2 0 2 で受信したレスポンスデータから、レスポンスを生成したデータ生成器 1 0 0 の識別子を取り出し、鍵生成用データ記憶部 2 0 9 に記憶する。

【ステップ S 1 5 2】：レスポンスデータに暗号化された前鍵が含まれていれば、それを取り出し、前鍵復号部 2 0 8 で復号し、その結果を前鍵記憶部 2 0 7 に記憶する。もし、暗号化された前鍵が含まれていなければ、何もせずステップ S 1 5 3 へ進む。

【ステップ S153】：鍵生成部 206 で、前鍵記憶部 207 に記憶されている前鍵と、鍵生成用データ記憶部 209 に記憶されている認証実行日時とデータ生成器 100 の識別子から、鍵を生成する。

【ステップ S154】：レスポンスデータから暗号化されたレスポンスを取り出し、鍵生成部 206 で生成された鍵を使用して、復号部 204 で復号する。

【ステップ S155】：復号部 204 での復号の結果が、チャレンジ記憶部 222 に記憶されているチャレンジおよびコミットメント記憶部 220 に記憶されているコミットメントに照らして正しいものがあるかどうかを検査する。

【0087】

データ検証器 200 が、ステップ S155 の検査で、正しいレスポンスあることが確認できた場合、データ検証器 200 にとっては、以下のことが保証される。

【0088】

1. データ生成器 100 が作成したレスポンスが、チャレンジおよびコミットメントに対して正しい署名であること。
2. データ検証器 200 が送付した認証実行日時が、差し替えられることなくデータ生成器 100 に届いたこと。
3. 受け取ったデータ生成器 100 の識別子が、レスポンスを生成したデータ生成器 100 と同一のものであること。

【0089】

【発明の効果】

以上説明したようにこの発明によれば主検査対象データの検証手続に関連して副検査対象の真正性を検査することができ、その際、ハッシュ値等の冗長データを付加する必要がない。

【図面の簡単な説明】

【図 1】 本発明の第 1 の実施例の構成を全体として示すブロック図である。

【図 2】 上述第 1 の実施例のデータ生成処理を説明するフローチャートである。

【図 3】 上述第 1 の実施例のデータ検証処理を説明するフローチャートである。

【図 4】 本発明の第 2 の実施例の構成を全体として示すブロック図である。

【図 5】 上述第 2 の実施例のデータ復号処理を説明するフローチャートである。

【図 6】 上述第 2 の実施例のデータ検証処理を説明するフローチャートである。

【図 7】 本発明の第 3 の実施例の構成を全体として示すブロック図である。

【図 8】 上述第 3 の実施例のレスポンス生成処理を説明するフローチャートである。

【図 9】 上述第 3 の実施例のレスポンス検証処理を説明するフローチャートである。

【図 1 0】 上述第 2 の実施例の変形例を説明する図である。

【符号の説明】

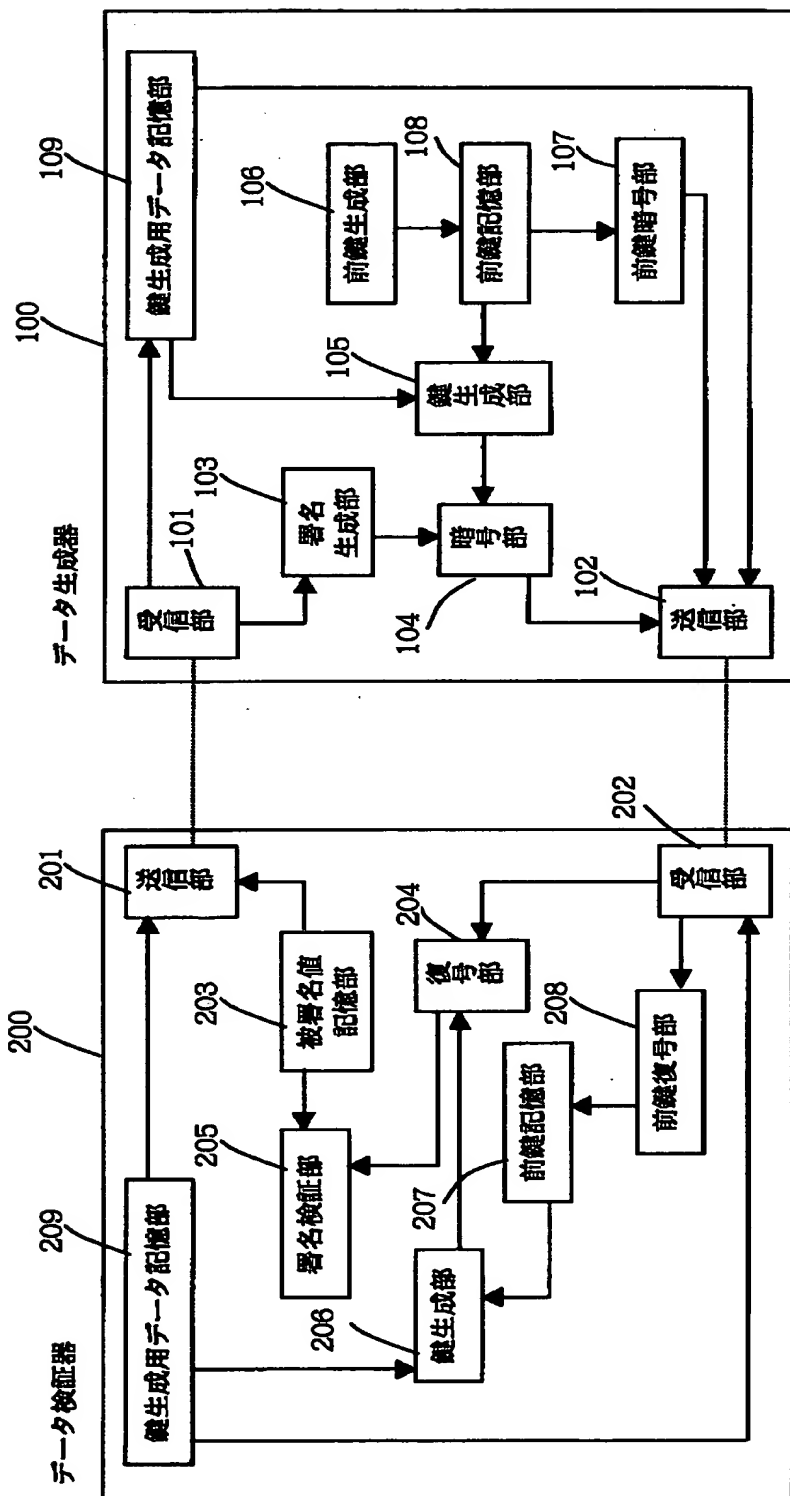
1 0 0	データ生成装置
1 0 1	受信部
1 0 2	送信部
1 0 3	署名生成部
1 0 4	暗号部
1 0 5	鍵生成部
1 0 6	前鍵生成部
1 0 7	前鍵暗号部
1 0 8	前鍵記憶部
1 0 9	鍵生成用データ記憶部
1 1 0	暗号文復号部
1 2 0	乱数生成部
1 2 1	乱数記憶部

1 2 2	コミットメント生成部
1 2 3	レスポンス生成部
2 0 0	データ検証器
2 0 1	送信部
2 0 2	受信部
2 0 3	被署名値記憶部
2 0 4	復号部
2 0 5	署名検証部
2 0 6	鍵生成部
2 0 7	前鍵記憶部
2 0 8	前鍵復号部
2 0 9	鍵生成用データ記憶部
2 1 0	暗号文記憶部
2 1 1	復号結果検証部
2 2 0	コミットメント記憶部
2 2 1	レスポンス検証部
2 2 2	チャレンジ記憶部

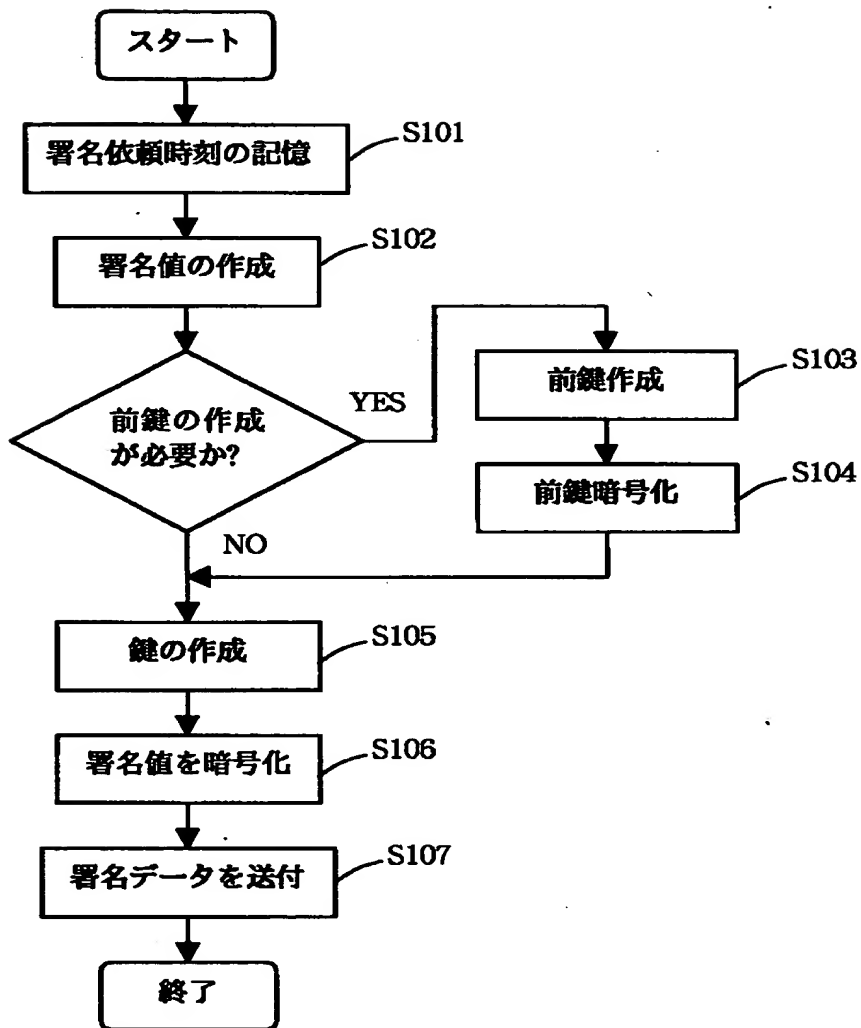
【書類名】

図面

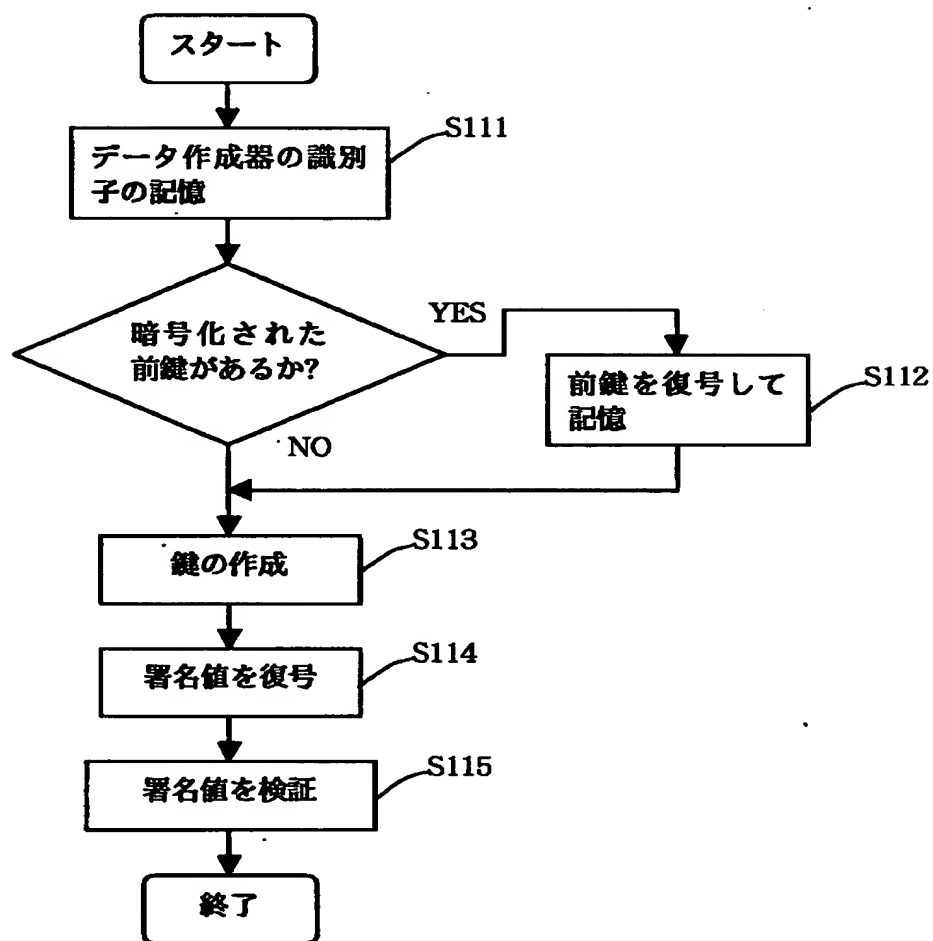
【図 1】



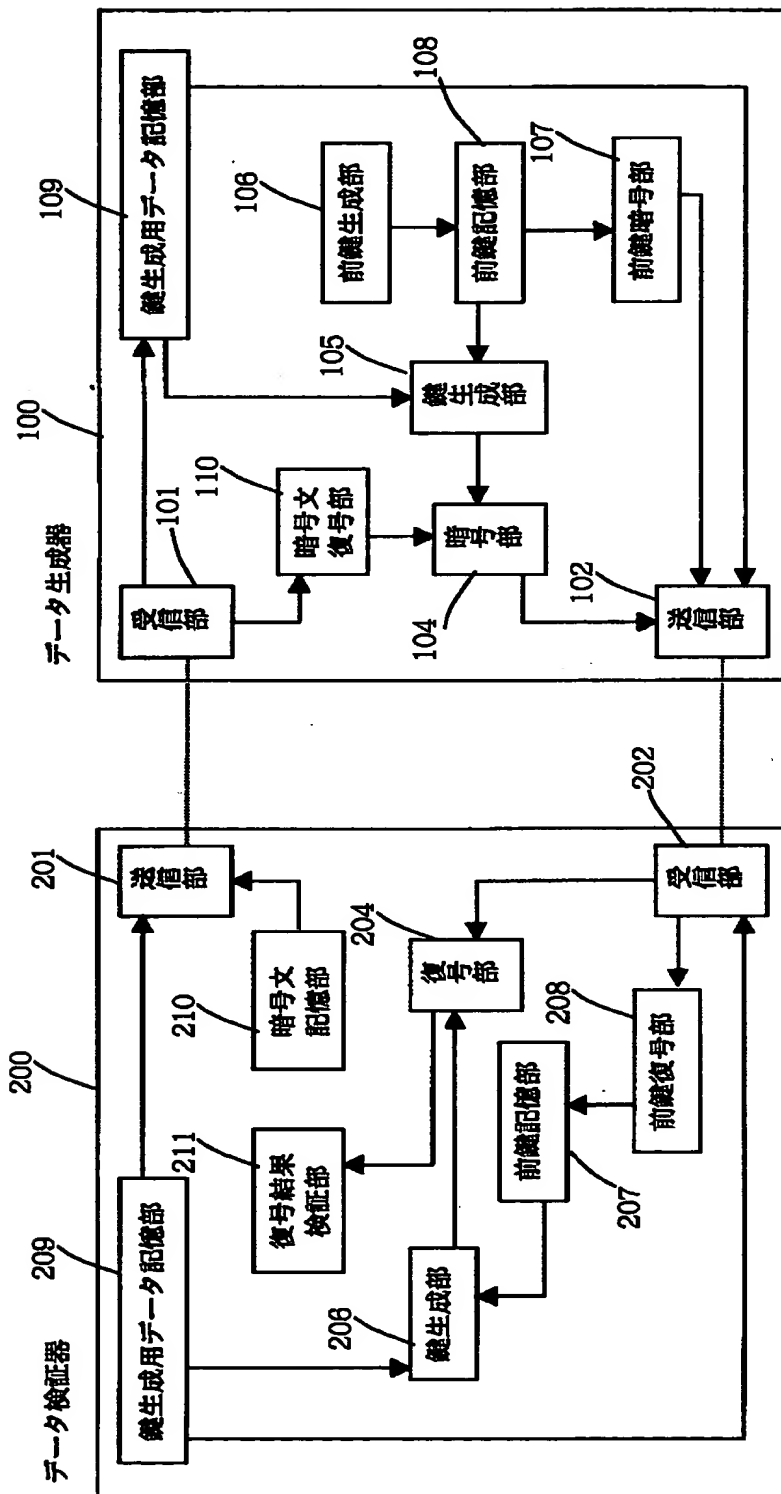
【図 2】



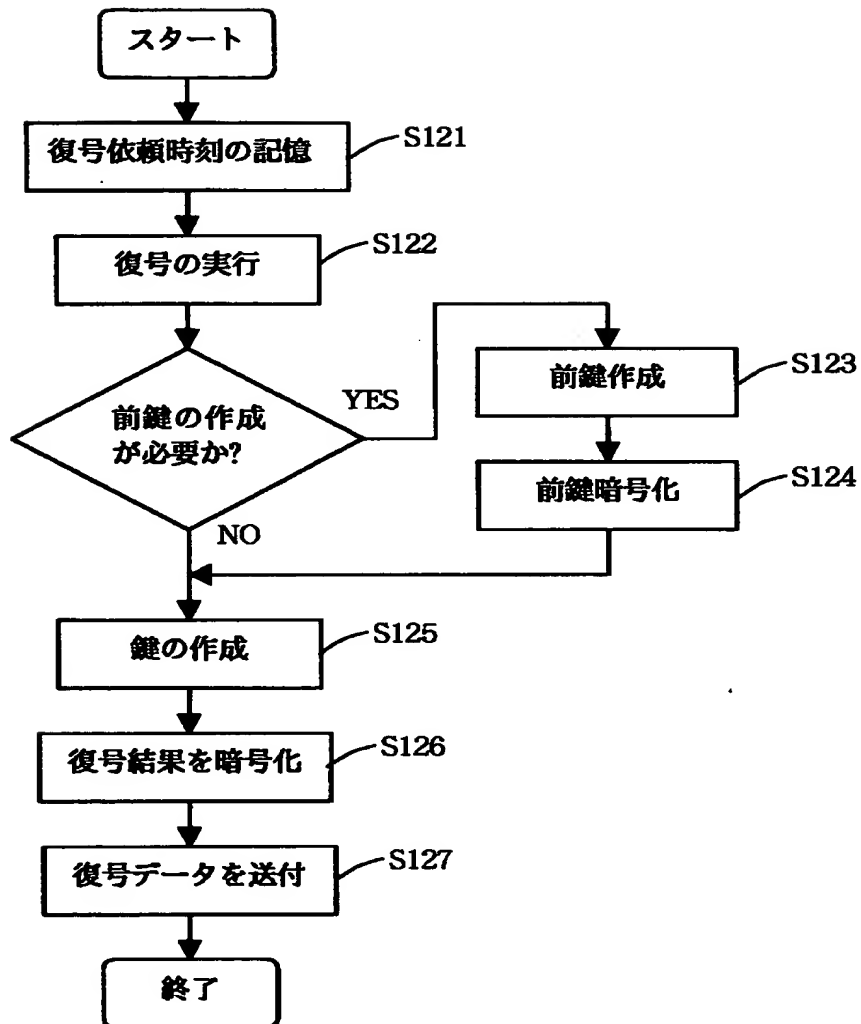
【図 3】



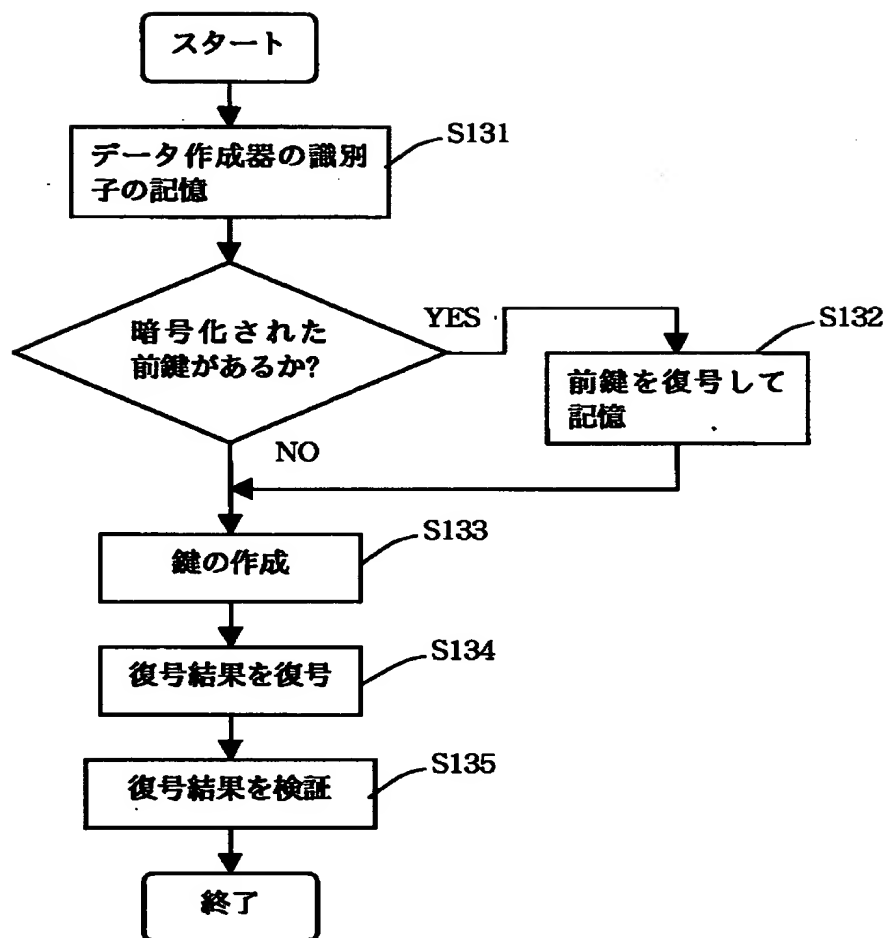
【図 4】



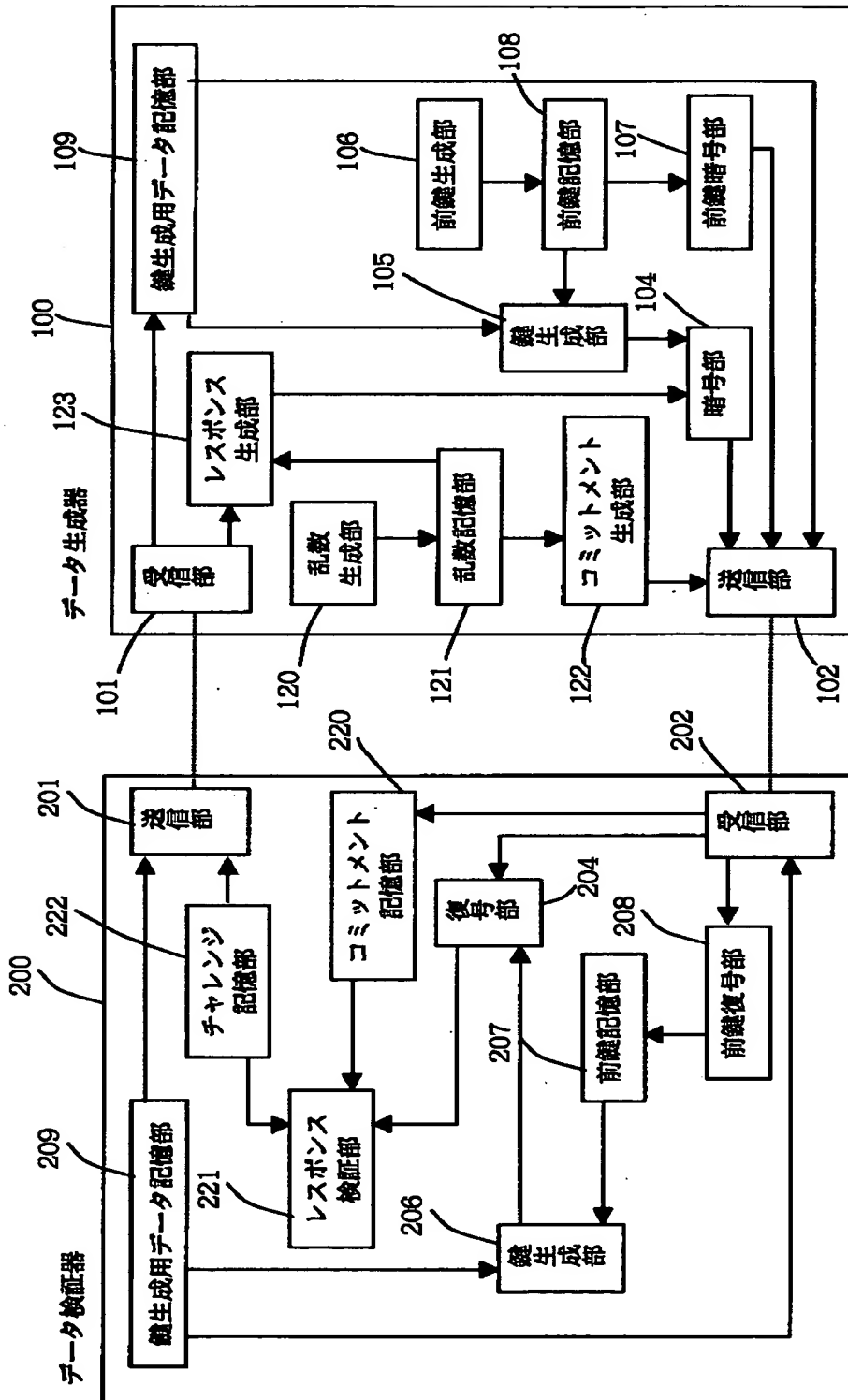
【図 5】



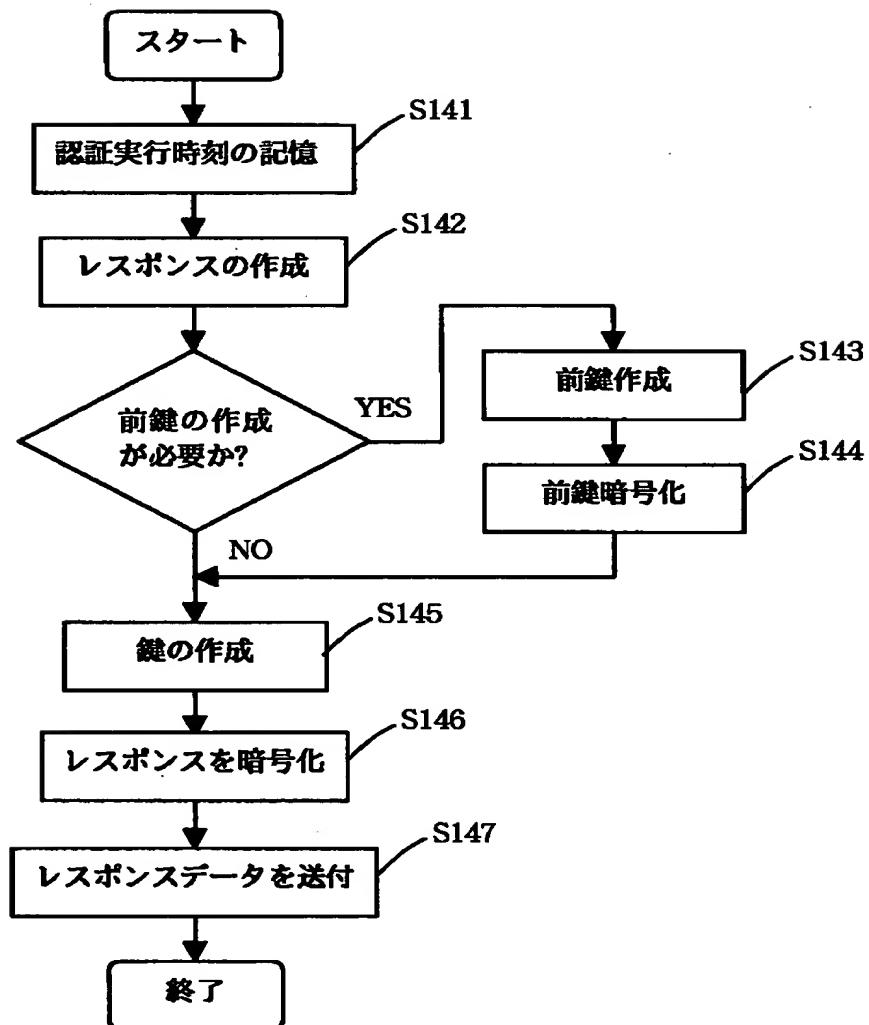
【図 6】



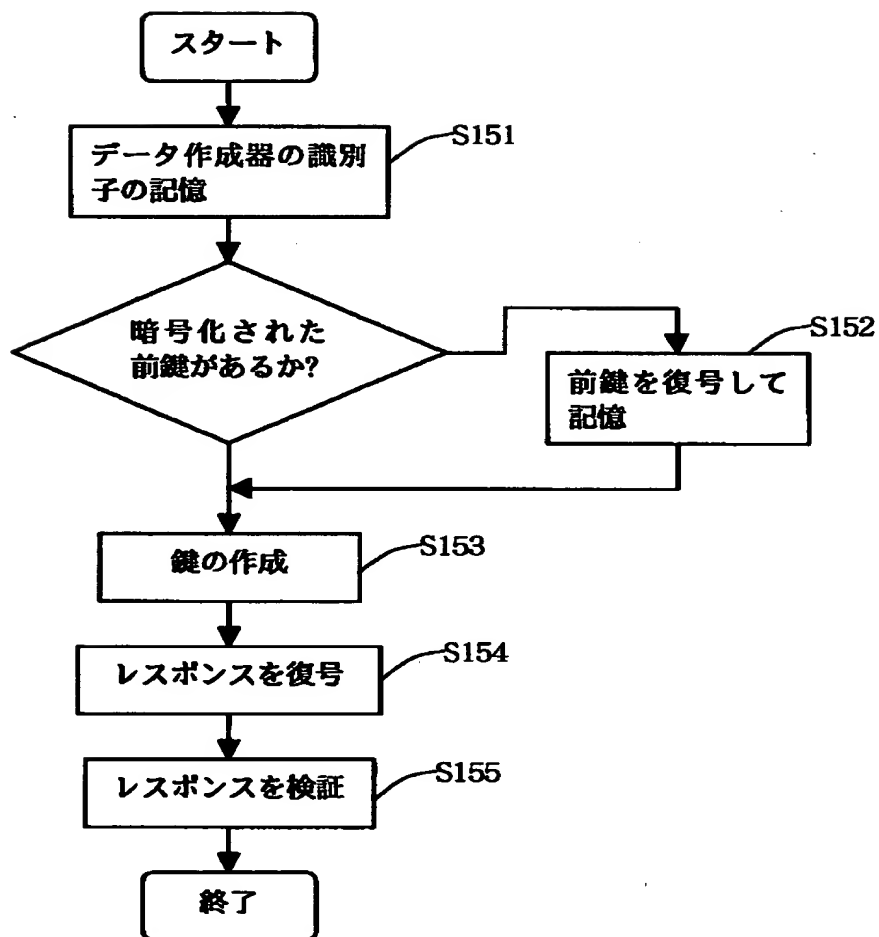
【図 7】



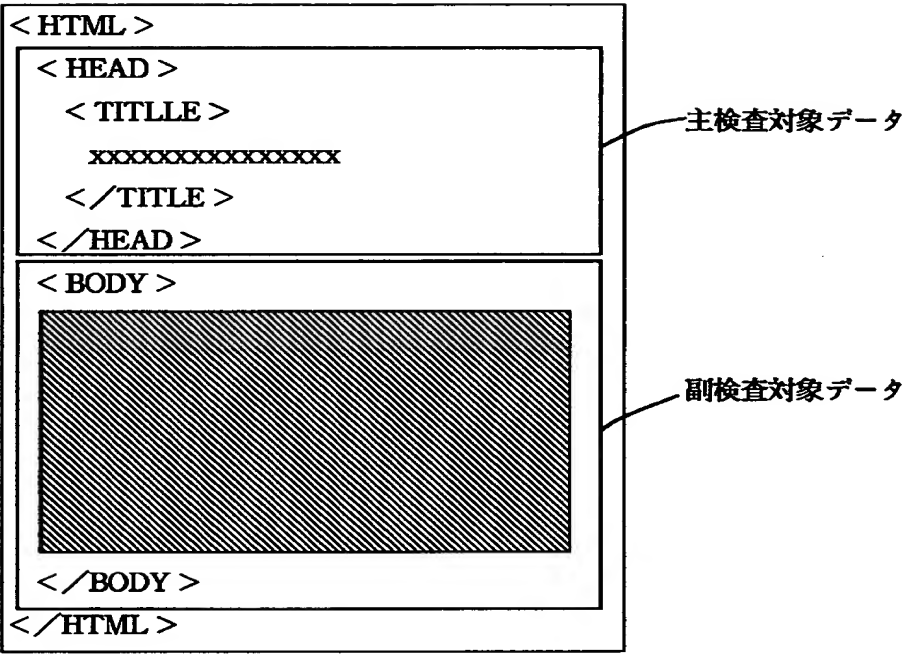
【図 8】



【図 9】



【図 1 0】



【書類名】 要約書

【要約】

【課題】 データの完全性の保証を簡易に行う。

【解決手段】 データ検証器 2 0 0 が被署名値と署名依頼時刻とから署名作成依頼を作成し、データ生成器 1 0 0 に送付する。データ生成器 1 0 0 が、署名作成依頼から、被署名値を取り出し、その署名生成部 1 0 3 が、被署名値に対する署名値を作成する。鍵生成部 1 0 5 が、前鍵と、署名作成依頼中の署名依頼時刻と、データ生成器 1 0 0 の識別子から、鍵を生成し、暗号部 1 0 4 が署名値を暗号化する。この後、暗号化署名値等をデータ検証器 2 0 0 に返信する。データ検証器 2 0 0 は、鍵生成部 2 0 6 で、前鍵と、署名依頼時刻と、データ生成器 1 0 0 の識別子とから、復号鍵を生成し、さらに、暗号化署名値を取り出し、復号部 2 0 4 で復号する。署名検証部 2 0 5 が復号の結果が正しい署名であるかどうかを検査する。

【選択図】 図 1

出 願 人 履 歴 情 報

識別番号 [000005496]

1. 変更年月日 1996年 5月29日

[変更理由] 住所変更

住 所 東京都港区赤坂二丁目17番22号

氏 名 富士ゼロックス株式会社